Module 1 : JavaScript

Introduction

JavaScript:

- JavaScript is a programming language which allows us add interactivity to web pages created in 1995.
- JavaScript can run on
  - client side on web page in web browser and
  - also on web servers to access database & file system.

Comments:

- Comments in JS language are used for documenting code and also to prevent code execution of particular line/lines
- There are 2 types of comments in the JS language.
  - Single Line Comments
    - Using: //
  - Multi-Line Comments
    - Using: /* … */

Input & Output:

- For Command Line I/O JS has following options:
- **Input:**
  - Module called prompt-sync
- **Output:**
  - console.log()
  - process.stdout().write()

Fundamentals

Keywords:

var	delete	for	let	break
super	void	case	do	static
function	new	switch	while	interface
catch	else	if	package	finally
this	with	class	enum	default
implements	private	throw	yield	typeof
const	export	import	protected	return
true	continue	extends	in	instanceof
public	try	debugger	false	

Data Types:

In JavaScript there are 5 different data types that can contain values:

- string
- number
- boolean
- object
- function

There are 3 types of objects:

- Object
- Date
- Array

Variable & Constants

Variable

- let keyword is used to declare a variable
- Rules:
  - Variables defined with let cannot be Redeclared.
  - Variables defined with let must be declared before use.
  - Variables defined with let have Block Scope.

Constant:

- Const keyword is used to declare a constant.
- Rules:
  - Variable must be assigned a value when they are declared.
  - Variables defined with const cannot be Redeclared.
  - Variables defined with const cannot be Reassigned.
  - Variables defined with const have Block Scope.

Rules for Naming:

- A variable can have alphabets, digits, $ sign and underscore.
- A variable name can start with the alphabet, and underscore only. It can't start with a digit.
- No whitespace is allowed within the variable name.
- Names are case sensitive.
- A variable name must not be any reserved word or keyword, e.g. int, float, etc.

Operators

Arithmetic Operators

Operators	Meaning	Example	Result
+	Addition	4+2	6
-	Subtraction	4-2	2
*	Multiplication	4*2	8
/	Division	4/2	2
%	Modulus operator to get remainder in integer division	5%2	1
++	Increment	A = 10; A++	11
--	Decrement	A = 10; A--	9

Assignment Operators

Operator	Example	Equivalent Expression
=	$m = 10$	$m = 10$
+=	$m += 10$	$m = m + 10$
-=	$m -= 10$	$m = m - 10$
*=	$m *= 10$	$m = m * 10$
/=	$m /=$	$m = m/10$
%=	$m \% = 10$	$m = m\%10$

Relational Operators

Operators	Meaning	Example	Result
<	Less than	5<2	False
>	Greater than	5>2	True
<=	Less than or equal to	5<=2	False
>=	Greater than or equal to	5>=2	True
==	Equal to	5==2	False
! =	Not equal to	5! =2	True
===	Equal value and same type	5 === 5	True
		5 === "5"	False
! ==	Not Equal value or Not same type	5 ! == 5	False
		5 ! == "5"	True

Logical Operators

Operator	Meaning	Example	Result
&&	Logical and	(5<2)&&(5>3)	False
||	Logical or	(5<2)||(5>3)	True
!	Logical not	!(5<2)	True

Control Statements

Selection Statements

For Selection Statements JS provides the following options:

- if statement
- if else statement
- if else if statement
- switch case

Iteration Statements

For iteration Statements JS provides the following options:

- while loop
- do while loop
- for loop
- for in loop
- for of loop

Arrays

Arrays:

- array is collection of elements of any type.
- array is represented by square brackets [] and elements separated by comma (,).
- The size of array is not fixed.
- To access array elements we can use array_name[index].
- To find the number of elements we use array_name.length.

Array methods to add & remove:

Add elements:

- push() : it adds an element at the end of the array.
- unshift(): it adds an element at the beginning of the array.

Remove Elements:

- pop(): it removes an element from the end of the array.
- shift(): it removes element from the beginning of array.

Note: splice() method can be used to add, update and remove elements.

More Array methods:

- sort(): it sorts the elements in the ascending order.
- reverse(): it reverses the elements in the array.
- concat(): it merges one or more arrays and returns a merged array.
- join(): it joins all the elements of the array using a separator(,) and returns a string.

Strings

Strings:

- String is created by surrounding them in quotes.
- For single line string we can use single and double quotes.
- For single and multiline strings we can use backticks.
- Backticks can be used to include variables or expressions into a string.

Working with String:

- To find the length of string we can use length property.
- To access string characters:
  - stringname[index]
  - stringname.charAt(index)

String methods:

- concat(): joins two or more strings.
- replace(): it replaces a string with another string.
- substring(): it returns a part of string.
- trim(): is removes whitespaces from the string.
- toLowerCase(): it returns the passed string in lowercase.
- toUpperCase(): it returns the passed string in uppercase.

Functions

Functions:

- It is a block of code that performs a specific task.
- They make the code easy to understand and reusable.
- JS has two types of functions:
  - Built-in functions
  - User defined functions.

Function Definition & Call:

- To declare user defined function:
  - function function_name() {
    - Function body
  - }
- To call the function:
  - function_name()

Function expressions:

- Function can be defined as expressions.
  - let var_name = function(par) { return value; }
- To call the function we use:
  - let new_var_name = var_name(par)
- The above is also called an anonymous function.

Arrow functions:

- Arrow functions allows us to create functions in a cleaner way compared to regular functions.
- Syntax is:
  - let function_name = (args) => {
  - statements(s)
  - }

Object Oriented Programming

Object Oriented Prog:

- Classes:
  - Class is a blueprint of an Object. It defines the states/variable and behavior/methods.
  - Process of creating class is called encapsulation.
- Objects:
  - Object is an instance of class.
  - Process of creating object is called instantiation.

Class and Object:

- To define class:
  - class className {
    - constructor(parameter_list) {
    - var_declaration
  - }
  - method1(par) {
    - method body
  - }
  - }

- To create object:
  - Let o1 = new className(parameter_list)

constructor:

- It is a special method which is called when object I created.
- It is useful for initialization of variables/state/properties of class.

Inheritance:

- When one class derived the properties and methods of another class it is called inheritance in OOP.
- The class that inherits the property is known as subclass or child class and the class whose properties are inherited is known as a superclass or parent class.
- The main advantage of inheritance is reusability.

HTML

HTML:

- HTML stands for Hyper Text Markup Language, most widely used to develop web pages.
- HTML was created by Berners-Lee in 1991 and HTML-5.x is the latest version.
- Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.
- Markup Language means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.

HTML Page Structure:

```
<html>
  <head>
    <title>Page title</title>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
    <p>This is another paragraph.</p>
  </body>
</html>
```

HTML Tags:

- HTML uses various tags to format the content.
- HTML tags are not case sensitive, but its recommended to write in lowercase.
- These tags are enclosed with angle braces <Tag Name>.
- Mostly tags has opening tag and closing tag.
- <html> is opening tag and </html> is closing tag.
- Never skip the closing tag else we can get unexpected results.
- The purpose of web browser is to read HTML documents and display them correctly. Browser does not display the HTML tags, but uses them to determine how to display the document.

HTML Element:

- An HTML element is defined by:
  - Start tag
  - Some content
  - End tag
- Format:
- <tagname> Content </tagname>
- Eg:
  - <h1>This is a heading</h1>
  - Hello World!

CSS

Cascading Style Sheets:

- CSS is used to format the layout of a webpage.
- With CSS we can control:
  - color, font, text size, etc
- CSS can be added to the HTML documents in 3 ways:
  - **Inline:**
    - by using the <style> attribute inside HTML elements.
  - **Internal:**
    - by using a <style> element in the <head> section.
  - **External:**
    - by using a <link> element to link to an external CSS file.

Precedence of style rules:

- The type of style being considered more important than other is known as precedence of style.
- Order of precedence is as follow:
  - Inline style
  - Internal style
  - External style
  - Browser default style

Selector types:

- To apply the style to an element or more than one element we use selectors.
- Few types of selectors are: tag selector, class selector , id selector etc

Tag Selector:

- To select tags by their name and apply styles to them we use tag selector.
- Syntax is:
  - **tagname {**
    declaration_list;
  - **}**

id Selector:

- To select tags by their id attribute value and apply styles to them we use id selector.
- Syntax is:
  - **#idattributevalue {**
    declaration_list;
  - **}**

class Selector:

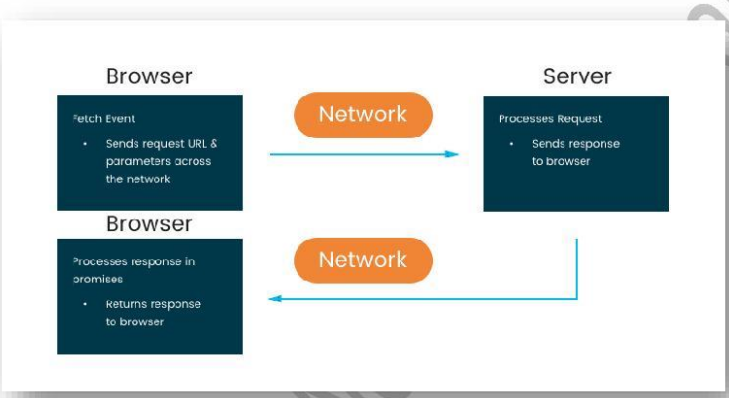
- To select tags by their class attribute value and apply styles to them we use class selector.
- Syntax is:
  - **.classattributevalue {**
    declaration_list;
  - **}**
- We can specify the same class attribute value to more than one tag.
- An html element can have list of class attribute values separated by white space.

fetch

fetch API:

- fetch API allows us to make HTTP requests to the web servers.
- fetch() requires only one parameter which is the URL to the resource we want to fetch
- fetch() returns a Promise(it is an object that encapsulates the result of asynchronous operation) so we can use then() and catch()

fetch API Working:



1. We send request using fetch(url) to the web browser.
2. Server processes the request.
3. If the request is fulfilled, Promise will resolve in Response object and using response we can get the data.
4. If request is rejected (for reasons such as timeout, loss of network) Promise will resolve into Error object.